



Olimpíada Brasileira de Informática

OBI2026

Caderno de Tarefas

Modalidade Programação • Nível Sênior • Fase 1 - Turno B

18 a 19 de Junho de 2026

A PROVA TEM DURAÇÃO DE 2 horas

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 11 páginas (não contando a folha de rosto), numeradas de 1 a 11. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como “Digite o dado de entrada:” ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Você pode submeter até 50 soluções para cada tarefa. A pontuação total de cada tarefa é a melhor pontuação entre todas as submissões. Se a tarefa tem sub-tarefas, para cada sub-tarefa é considerada a melhor pontuação entre todas as submissões.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
 - em C: *scanf*, *getchar*, *printf*, *putchar*;
 - em C++: as mesmas de C ou os objetos *cout* e *cin*.
 - em Java: qualquer classe ou função padrão, como por exemplo *Scanner*, *BufferedReader*, *BufferedWriter* e *System.out.println*
 - em Python: *read*, *readline*, *readlines*, *input*, *print*, *write*
 - em Javascript: *scanf*, *printf*
- Procure resolver a tarefa de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Dedos

Nome do arquivo: `dedos.c`, `dedos.cpp`, `dedos.pas`, `dedos.java`, `dedos.js` ou `dedos.py`

Rafaela descobriu uma forma muito interessante de representar números usando os dedos da mão. Antes, ela representava os números de 0 a 5 de acordo com a quantidade de dedos levantados, independente de quais eram esses dedos. Agora, ela aprendeu que é possível representar os números de 0 a 31 da seguinte forma:

- Se o dedo mínimo estiver levantado, ela deve somar 16 ao resultado.
- Se o dedo anelar estiver levantado, ela deve somar 8 ao resultado.
- Se o dedo médio estiver levantado, ela deve somar 4 ao resultado.
- Se o dedo indicador estiver levantado, ela deve somar 2 ao resultado.
- Se o dedo polegar estiver levantado, ela deve somar 1 ao resultado.

Por exemplo, se os dedos médio e indicador estiverem levantados (e os outros abaixados), conforma a imagem:



Neste sistema representa o valor $4 + 2 = 6$.

A garota está muito animada com a descoberta e agora deseja aprimorar suas habilidades em reconhecer os números representados nesse sistema. Assim, dados os valores D_1, D_2, D_3, D_4 e D_5 , que representam os dedos do mínimo ao polegar, calcule qual número Rafaela está indicando. Se $D_i = 1$, o i -ésimo dedo está levantado e, se $D_i = 0$, ele está abaixado ($1 \leq i \leq 5$).

Entrada

A primeira linha da entrada contém um inteiro D_1 , que representa o dedo mínimo, ou seja, se Rafaela deve somar ou não o valor 16.

A segunda linha da entrada contém um inteiro D_2 , que representa o dedo anelar, ou seja, se Rafaela deve somar ou não o valor 8.

A terceira linha da entrada contém um inteiro D_3 , que representa o dedo médio, ou seja, se Rafaela deve somar ou não o valor 4.

A quarta linha da entrada contém um inteiro D_4 , que representa o dedo indicador, ou seja, se Rafaela deve somar ou não o valor 2.

A quinta linha da entrada contém um inteiro D_5 , que representa o dedo polegar, ou seja, se Rafaela deve somar ou não o valor 1.

Saída

A saída deve conter um único inteiro, o número indicado por Rafaela. Não coloque nada além disso na saída, senão será considerado errado.

Restrições

- $0 \leq D_i \leq 1$, para todo $1 \leq i \leq 5$.

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (100 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
0 0 1 1 0	6

Explicação do exemplo 1: Note que esse é o exemplo presente no enunciado.

- Como o dedo médio está levantado, Rafaela deve somar 4 ao resultado.
- Como o dedo indicador está levantado, Rafaela deve somar 2 ao resultado.

Logo, o resultado é $4 + 2 = 6$. Perceba que a saída deve conter apenas o número 6, nada além disso.

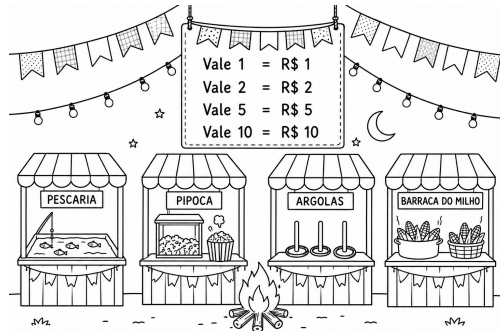
Exemplo de entrada 2	Exemplo de saída 2
0 0 0 0 0	0

Exemplo de entrada 3	Exemplo de saída 3
1 1 1 1 1	31

Exemplo de entrada 4	Exemplo de saída 4
1 0 1 0 1	21

Fichas

Nome do arquivo: `fichas.c`, `fichas.cpp`, `fichas.pas`, `fichas.java`, `fichas.js` ou `fichas.py`



Gustavo está muito animado para ir à festa junina de seu bairro. Para ele, essa é a melhor festa do ano, por causa das deliciosas comidas típicas, como milho, canjica e arroz-doce, e também das brincadeiras e atividades presentes na festa, como a pescaria e o jogo de argolas.

No dia da festa, a mãe do garoto lhe deu X reais para gastar no evento. No entanto, para organizar e facilitar os pagamentos internos, a festa junina do bairro de Gustavo usa um sistema de fichas, as quais podem ser de quatro valores: R\$1, R\$2, R\$5 e R\$10. Dessa forma, ao chegar na festa, ele precisa trocar seu dinheiro por fichas para pagar pelas comidas e brincadeiras.

Gustavo tem certeza que vai gastar tudo que recebeu, mas não quer carregar muitas fichas. Portanto, dada a quantidade X de reais que o garoto ganhou da mãe, ajude-o a calcular qual o mínimo de fichas pelas quais ele pode trocar o dinheiro.

Entrada

A entrada contém uma única linha com um inteiro X , o dinheiro de Gustavo.

Saída

A saída deve conter um único inteiro, o mínimo de fichas que Gustavo precisa para trocar todo seu dinheiro.

Restrições

- $1 \leq X \leq 200$.

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (20 pontos):** $X \leq 9$.
- **Subtarefa 2 (20 pontos):** X é múltiplo de 5.
- **Subtarefa 2 (60 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
8	3

Explicação do exemplo 1: Nesse caso, são necessárias 3 fichas:

- 1 ficha de R\$1.
- 1 ficha de R\$2.
- 1 ficha de R\$5.

Note que esse exemplo satisfaz as restrições da Subtarefa 2.

Exemplo de entrada 2	Exemplo de saída 2
75	8

Explicação do exemplo 2: Nesse caso, são necessárias 8 fichas:

- 1 ficha de R\$5.
- 7 fichas de R\$10.

Note que esse exemplo satisfaz as restrições da Subtarefa 3.

Exemplo de entrada 3	Exemplo de saída 3
199	22

Explicação do exemplo 3: Nesse caso, são necessárias 22 fichas:

- 2 fichas de R\$2.
- 1 ficha de R\$5.
- 19 fichas de R\$10.

Transporte

Nome do arquivo: `transporte.c`, `transporte.cpp`, `transporte.pas`, `transporte.java`,
`transporte.js` ou `transporte.py`

A semana olímpica é muito aguardada e almejada por todos os competidores da OBI. Esse evento é muito proveitoso para os alunos, devido não só às aulas e aos simulados de alto nível, mas também às amizades formadas durante o evento. No entanto, como alunos de todo o país se reúnem na competição, alguns precisam de transporte do aeroporto para o hotel no dia da chegada.

Todo ano, alguns ônibus são contratados para realizar esse deslocamento, mas a logística desse processo é um pouco complicada. Isso porque cada um dos N competidores chega em um horário, sendo que o i -ésimo deles chega no minuto t_i do dia ($1 \leq i \leq N$). Além disso, para não preocupar os pais e alunos, a OBI garante que nenhum aluno precisará esperar mais de T minutos desde o momento do pouso até a saída de seu transporte.

Os ônibus contratados são grandes o suficiente para levarem até N passageiros, porém, como os alunos não devem esperar por muito tempo para serem levados, podem ser necessários muitos transportes para levar todos do aeroporto para o hotel. Assim, dados a quantidade N de competidores, o tempo máximo de espera T e os tempos de chegada t_1, t_2, \dots, t_N , ajude a OBI a calcular a quantidade mínima de ônibus que ela precisa contratar para transportar todos os competidores.

Entrada

A primeira linha da entrada contém dois inteiros, N e T , respectivamente o número de competidores e o tempo máximo de espera. A segunda linha da entrada contém N números inteiros t_1, t_2, \dots, t_N , os tempos de chegada dos competidores.

Saída

A saída deve conter um único inteiro, a quantidade mínima de ônibus que a OBI precisa contratar para transportar todos os competidores.

Restrições

- $3 \leq N \leq 100$.
- $0 \leq T \leq 1000$.
- $1 \leq t_1 \leq t_2 \leq \dots \leq t_N \leq 1000$.

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (20 pontos):** $N = 3$.
- **Subtarefa 3 (30 pontos):** $T = 0$.
- **Subtarefa 4 (50 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 8 10 5 8 15 20 21 25 42 50	Exemplo de saída 1 3
--	--------------------------------

Explicação do exemplo 1: Nesse caso, são necessários 3 ônibus para levar todos os alunos:

- O primeiro pode sair no minuto 8, levando os alunos 1 (com tempo de espera de 3 minutos) e 2 (com tempo de espera de 0 minuto).
- O segundo, pode sair no minuto 25, levando os alunos 3, 4, 5 e 6, perceba que, em relação a este ônibus, o aluno com maior tempo de espera será o aluno 3 com 10 minutos de espera.
- Por fim, o terceiro pode sair no minuto 52, levando os alunos 7 (com tempo de espera de 10 minutos) e 8 (com tempo de espera de 2 minutos).

Perceba que, desta forma, nenhum aluno precisou esperar mais de $T = 10$ minutos.

Exemplo de entrada 2 3 10 1 11 12	Exemplo de saída 2 2
--	--------------------------------

Explicação do exemplo 2: Perceba que este exemplo satisfaz as restrições da subtarefa 2.

Exemplo de entrada 3 5 0 1 1 2 2 3	Exemplo de saída 3 3
---	--------------------------------

Explicação do exemplo 3: Perceba que este exemplo satisfaz as restrições da subtarefa 3.

Resumo de Números

Nome do arquivo: `resumo.c`, `resumo.cpp`, `resumo.pas`, `resumo.java`, `resumo.js` ou `resumo.py`

Ricardinho é um garoto prodígio e, após ganhar uma caixa com blocos numerados de 0 a 9 de seus pais, aprendeu a contar até 999 999 999. Além disso, as habilidades matemáticas do menino ficam cada vez melhores, e agora ele também sabe o que são números pares e ímpares.

Ele deseja testar seu aprendizado, mas, como toda criança, Ricardinho se cansou rapidamente de sua última brincadeira, a torre de números. Por isso, ele inventou um novo jogo, o resumo de números! Para resumir um número M formado por até 9 blocos, Ricardinho realiza os seguintes passos:

- Primeiro, ele pega um bloco T na caixa, cujo valor é igual ao total de blocos que compõe M .
- Em seguida, ele pega um bloco I na caixa, cujo valor é igual ao total de blocos **ímpares** que compõe M .
- Além disso, ele pega um bloco P na caixa, cujo valor é igual ao total de blocos **pares** que compõe M .
- Por fim, Ricardinho diz que o resumo de M é o número formado pelos blocos T, I e P , nesta ordem.

O garoto gostou muito de seu novo jogo, mas percebeu que alguns números são **chatos**. Ricardinho acha que um número é **chato** se ele é o resumo de si mesmo. Agora, Ricardinho deseja saber quantas vezes um número N pode ser resumido até que ele se torne um número chato.

Por exemplo, para $N = 71$, a sequência de resumos feita por Ricardinho seria:

- 71 é formado pelos blocos 7 e 1. Assim, o garoto pegaria os blocos $T = 2, I = 2, P = 0$, formando o número 220.
- 220 é formado pelos blocos 2, 2 e 0. Assim, o garoto pegaria os blocos $T = 3, I = 0, P = 3$, formando o número 303.
- 303 é formado pelos blocos 3, 0 e 3. Assim, o garoto pegaria os blocos $T = 3, I = 2, P = 1$, formando o número 321.
- 321 é formado pelos blocos 3, 2 e 1. Assim, o garoto pegaria os blocos $T = 3, I = 2, P = 1$, formando o número 321. Ou seja, 321 é um número chato.

Portanto, seriam necessários 3 resumos até chegar em um número que é o resumo de si mesmo.

Ricardinho está muito animado com sua nova brincadeira, mas ainda é uma criança pequena e, novamente, está cansado depois de fazer tantas contas. Portanto, dado o número N , ajude o garoto a calcular quantas vezes esse valor pode ser resumido até que se torne um número chato.

Entrada

A entrada contém uma única linha com um único inteiro N , o número que Ricardinho deseja resumir.

Saída

A saída deve conter um único inteiro, a quantidade de vezes que N pode ser resumido até que se torne um número chato.

Restrições

É garantido que todo caso de teste satisfaz as restrições abaixo.

- $1 \leq N \leq 999\,999\,999$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (30 pontos):** $N = 2026$.
- **Subtarefa 3 (70 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
71	3

Explicação do exemplo 1: Note que este é o exemplo descrito no enunciado.

Exemplo de entrada 2	Exemplo de saída 2
321	0

Explicação do exemplo 2: Como 321 é o resumo de si mesmo, Ricardinho não precisa fazer nenhum resumo até chegar em um número chato.

Exemplo de entrada 3	Exemplo de saída 3
123456789	2

Explicação do exemplo 3: Neste caso, $N = 123\,456\,789$. Portanto, Ricardinho realizaria os seguintes passos:

- 123 456 789 é formado pelos blocos 1, 2, 3, 4, 5, 6, 7, 8 e 9. Assim, o garoto pegaria os blocos $T = 9, I = 5, P = 4$, formando o número 954.
- 954 é formado pelos blocos 9, 5 e 4. Assim, o garoto pegaria os blocos $T = 3, I = 2, P = 1$, formando o número 321.
- 321 é formado pelos blocos 3, 2 e 1. Assim, o garoto pegaria os blocos $T = 3, I = 2, P = 1$, formando o número 321. Ou seja, 321 é um número chato.

Portanto, ele faria 2 resumos até chegar em um número chato.

